

Санкт-Петербургский государственный университет

Прикладная математика и информатика
Нелинейная динамика, информатика и управление

Трапезникова Евгения Викторовна

Разработка системы поддержки логистики на базе мультиагентных технологий

Бакалаврская работа

Научный руководитель:
к. ф.-м. н., доцент Кияев В.И.

Рецензент:
аспирант СПбГУ Иванский Ю.В.

Санкт-Петербург
2016

SAINT-PETERSBURG STATE UNIVERSITY

Applied Mathematics and Computer Science
Nonlinear Dynamics, Computer Science and Control

Evgeniia Trapeznikova

Development of logistics support system based on multi-agent technology

Bachelor's Thesis

Scientific supervisor:
Associate professor Vladimir Kiyaev

Reviewer:
Postgraduate student Yuriy Iwanskiy

Saint-Petersburg
2016

Оглавление

Введение	5
1. Мультиагентные системы	8
1.1. Мультиагентный подход	8
1.2. Понятие агента	10
1.3. Многоагентная система	13
1.4. Современные международные стандарты создания агентов и платформы МАС	14
1.5. Выводы по первой главе	15
2. Использование мультиагентного подхода для решения задач планирования	17
2.1. Некоторые сведения из логистики	17
2.2. Постановка задачи адаптивного планирования	18
2.3. Концепция сетей потребностей и возможностей (ПВ-сеть)	20
2.4. Модель организации перевозок	24
2.5. Процесс планирования	26
2.6. Модельный пример расчета прибыли	27
2.7. Выводы по второй главе	29
3. Разработка мультиагентной системы для задачи грузоперевозки	31
3.1. Общее описание системы	31
3.2. Агент «Погода вдоль трассы»	33
3.3. Агент «Оптимальная скорость»	35
3.4. Агент «Погрузка/разгрузка»	36
3.5. Агент «Самочувствие водителя»	38
3.6. Агент «Техническое состояние автомобиля»	39
3.7. Процесс мониторинга и архитектура уровня приложений и инфраструктуры	43
3.8. Создание интерфейса для реализации системы мониторинга и управления грузоперевозками	44

3.8.1. Инструментальные средства	44
3.8.2. Описание структуры базы данных и схема процес- са грузоперевозки	45
3.8.3. Пример пользовательского интерфейса	48
3.9. Выводы по третьей главе	53
Заключение	54
Список литературы	55

Введение

Экономическое благосостояние и развитие любой страны на современном этапе в большой степени зависит от её логистической инфраструктуры. Причем, чем больше территория страны, тем более важна логистическая составляющая в экономике. Отметим, что наряду с морским, речным, железнодорожным транспортом экономически развитые страны продолжают всемерно развивать сети автомобильных дорог. Опыт европейских стран, США, Японии и Южной Кореи подтверждают это. Это особенно хорошо видна на примере Соединенных Штатов Америки и Канады, где железные дороги используются, в основном, для перевозки горючего и крупногабаритных грузов, а основная часть грузов перевозят крупнотоннажными автомобилями. При этом эффективность доставки грузов в большой степени определяется вопросами планирования и точности реализации перевозок.

Площадь Российской Федерации составляет более 17 миллионов квадратных километров. В настоящее время полноценная автомобильная сеть покрывает европейскую часть территории России. Большая площадь означает необходимость большой и развитой сети дорог, покрывающей всю страну, а это означает всё возрастающее множество различных маршрутов передвижения. Будущее нашей страны в большой степени зависит не только от того, насколько успешно будет поддерживаться и развиваться эта сеть, но от эффективно организации доставки грузов. Именно это обстоятельство говорит об актуальности и важности выбранной темы.

Люди, передвигаясь из одного города в другой, обычно выбирают удобный и короткий путь. Однако, они всегда могут себе позволить в случае необходимости отойти от намеченного пути или выбрать более длинный маршрут. Но у организаций, занимающихся междугородними и международными грузоперевозками, часто нет такой возможности. Главная цель логистической компании: совершить перевозку с максимально возможной точностью, что позволит получить хорошую прибыль. Для этого нужно разумно выбирать максимально удобный

(оптимальный) маршрут и передвигаться по нему согласно запланированному расписанию.

Планирование маршрута движения грузовиков должно быть максимально рационально и учитывать различные факторы. Во-первых, необходимо рассчитывать время перевозок исходя из состояния дороги, по которой будет проходить маршрут, а как нам все известно, качество дорог зачастую оставляет желать лучшего. Во-вторых, нельзя пренебрегать погодным фактором. Например, в зимнее время возможно сильное заснеживание трассы, что приведёт либо к её закрытию, либо к тому, что на дороге образуется скользкая грязевая масса, которая порядком снизит максимально возможную скорость движения. В случае закрытия трассы возникает задача перепланирования маршрута движения, причём оно должно быть проведено в кратчайшие сроки и, по возможности, максимально выгодно как для организации-подрядчика, так и для заказчика. В-третьих, нельзя забывать, что за рулём машины сидит живой человек, который утомляется, теряет концентрацию, отвлекается, что может в лучшем случае привести, например, к снижению скорости движения и отставанию от графика. Отметим и необходимость постоянно следить за состоянием силовых агрегатов многотонного автомобиля, тормозной системы, нагрузку на шасси и т. д.

Для того что бы быть конкурентоспособным, предприятие, занимающееся грузоперевозками, должно учитывать все указанные выше факторы, которые состоят из многих параметров. Вручную невозможно отследить и среагировать на все случившиеся или возможные неполадки, поэтому эффективная диспетчеризация с участием людей является довольно сложной деятельностью. Логично сказать, что система управления, занимающаяся планированием расписания и маршрутов доставки, должна быть максимально автоматизирована на базе современных методов планирования и управления сложными системами. В таком случае закономерен вопрос, как же будет выглядеть система, которая вначале должна составить расписание для доставки нового заказа (выбрать водителя, машину, которая будет перевозить заказ, построить маршрут исходя из желания максимизировать прибыль от

сделки и выполнить всё в срок), а потом, в случае незапланированных событий (невыполнение графика погрузки- разгрузки, поломка машины, вынужденное изменение маршрута) изменить уже имеющийся план перевозки с минимальными потерями.

С нашей точки зрения этой задачей справится система, построенная на основе мультиагентного подхода. Мультиагентная система представляет собой децентрализованную систему, позволяющую оперативно, гибко и эффективно практически самостоятельно принимать «разумные» решения, отражающие баланс интересов участников. Разработка и применение такого подхода в нашей работе для разработки означает содержание её научной новизны.

Таким образом, *предметной областью* нашего исследования является использование мультиагентных технологий в целях создания управляющих и мониторинговых систем.

Объектом исследования является современная логистическая компания, занимающаяся крупнотоннажными автомобильными перевозками, *предмет исследования* - система управления, построенная с использованием мультиагентных технологий.

Целью работы является разработка системы поддержки логистики на базе мультиагентных технологий.

Для достижения поставленной цели необходимо реализовать *следующие задачи*:

1. рассмотреть и проанализировать современное содержание мультиагентного подхода;
2. рассмотреть возможность применения мультиагентного подхода для задачи планирования;
3. разработать структуру мультиагентной системы для системы мониторинга;
4. разработать базу данных и создать интерфейс для реализации системы мониторинга и управления грузоперевозками.

1. Мультиагентные системы

1.1. Мультиагентный подход

Мультиагентные технологии в большой степени имеют своей основой природные явления и живые системы. Представим себе колонию муравьёв. Подавляющее большинство составляют муравьи-рабочие. Каждая такая отдельно взятая особь не обладает сколь-нибудь разумным интеллектом. Однако, если проследить за тем, как муравьи, например, ищут пищу и передают информацию от особи к особи, то появляется то, что позднее называли «квази-интеллектом» роя, стаи, сообщества животных. В естественной природе изначально все муравьи передвигаются спонтанно и при обнаружении в каком-либо месте пищи оставляют след из феромонов, двигаясь обратно в колонию. Если позже другие муравьи заметят этот след, то они будут стремиться к нему. Если они также находят пищу в том месте, откуда пришёл их предшественник, то они укрепляют феромонную тропу, возвращаясь домой. Если первый муравей нашёл большой источник пищи, то каждый представитель будет оставлять след феромонов по пути назад.

Со временем всё большее число муравьёв проходит по этому пути, поэтому он становится все более привлекательным (повышается «феромоновый потенциал») для их сородичей. Если в месте, куда ведёт тропа, больше не осталось пищи, то феромоны начнут испаряться, тем самым уменьшая привлекательность (потенциал) этого пути. На короткую тропу нужно меньшее время для прохождения. Поэтому плотность феромонов (значение потенциала) на коротких тропах будет выше, чем на длинных.

Эти взаимодействия муравьёв называются стигмергией. На них базируется поиск кратчайшего пути от колонии муравьёв до источника пищи. Это явление называется проявлением эмерджентного интеллекта, то есть наличие у системы свойств, не присущих её элементам.

Данный пример можно рассматривать как своеобразную мультиагентную систему природного мира.

Теперь же перенесёмся в область компьютерных технологий.

При использовании системы, работающей на базе мультиагентных технологий, нужно определить ключевую сущность, относительно которой рассматривается система. Такой вещью является понятие мобильного агента, работающий как независимая компьютерная программа.

Если для решения задачи используется традиционная схема построения программной среды, то для нахождения решения происходит поиск и построение определённого детерминированного алгоритма. В задачах с использованием мультиагентного подхода решение в некоторых обусловленных случаях генерируется автоматически в процессе взаимодействия множества программных агентов [2].

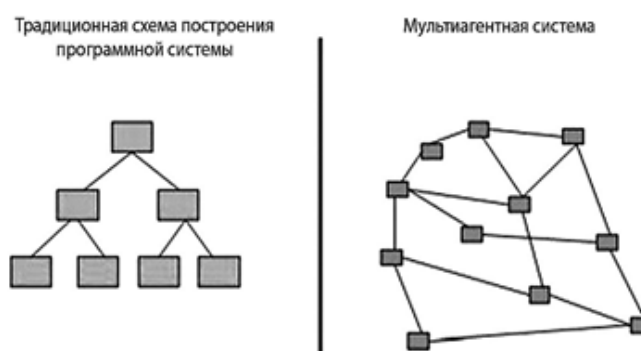


Рис. 1: Традиционное и мультиагентное построение продукта [12]

Уже сказано, что мультиагентным системам присуще проявление эмерджентности. Главная отличительная черта эмерджентного интеллекта - развитие и невозможность прогнозирования того, какие решения будут приняты. На практике приходится сталкиваться с большим числом взаимодействий агентов друг с другом, приводящее к нужному решению, которые в ручном режиме практически невозможно отследить. Но нам и не требуется отслеживание этих взаимодействий, поскольку агенты имеют свои цели, но не имеют строго определённых сценариев, согласно которым они должны достигаться.

Сценарии, согласно которым будут действовать агенты для реализации своих целей, создаются и исполняются самими агентами. На каждом этапе агенты анализируют входы управляющей системы и реагируют на незапланированные события (опоздания, неполадки, изменения).

Разные агенты имеют различное строение, поэтому и реакции на события отличаются, но все реакции можно разбить на два типа: самостоятельное реагирование или в команде с оператором (например, человеку отправляется сообщение о каких-либо неполадках). В итоге приходим к логическому выводу, что эмерджентный интеллект возникает не в результате специальной разработки, а как итог самоорганизации, возникающий в непредсказуемые моменты времени. После своего действия он так же неожиданно, как и возник, исчезает, но в процессе своего существования определяющим образом руководит работой всей системы.

1.2. Понятие агента

Понятие "агент" изменяется в зависимости от того контекста, где оно применяется. Поэтому не возможно дать устоявшееся определение этого слова. Однако, введём определение в рамках МАС. Удобнее и нагляднее будет это сделать через свойства, которыми агент должен обладать. *Интеллектуальный агент* – обособленная (самостоятельная) программная система, обладающая следующими свойствами [12]:

- *автономность* – независимость от действий, совершаемых владельцем, самостоятельное функционирование, самостоятельный контроль своих действий и мониторинг внутренних показателей;
- *адаптивность* – способность реагировать на постоянно происходящие изменения и способность обучаться;
- *коллаборативность* – исполнение агентом разных ролей по отношению к другим агентам в зависимости от ситуации;
- *способность к рассуждениям* – обладание агентом таких черт, как восприятие внешних данных и реагирование на внешние события, исходя из имеющихся сведений, заложенных в архитектуру агента;
- *коммуникативность* – способность агентов общаться друг с другом;

- *мобильность* – возможность пересылать код агента между контекстами (серверная среда);
- *социальное поведение* – общение с другими агентами и взаимодействие с ними, зависящее от результатов этих коммуникаций;
- *реактивность* – подходящая реакция на изменения среды и её восприятие;
- *активность* – способность создавать цели и совершать целесообразные действия;
- *наличие базовых знаний* – наличие у агента сведений о себе, среде, других агентах, которые не изменяются;
- *наличие убеждений* – часть знаний агента, которые могут меняться в течение взаимодействий агента в зависимости от текущих внутренних показателей или состояния среды;
- *наличие цели*;
- *наличие желаний* – состояния и/или ситуации, достижение которых для агента важно;
- *наличие обязательств* – задачи, возникающие перед агентом по просьбе и/или поручению других агентов;
- *наличие намерений* – те задачи, которые агенту необходимо сделать в зависимости от обязательств и желаний.

Дадим понятие *мультиагентной системы* (МАС), которое в свете рассмотренного выше является довольно очевидным: это такая система, в основе которой функционирует не менее двух агентов, обладающих способностью общаться между собой с целью разрешения имеющейся задачи. В МАС определено распределение задач среди всех агентов, каждый из которых предстаёт в роли члена группы, наделённого определённой мерой «ответственности» и требования к «опыту».

Базу *архитектуры* агента составляет контекст (серверная среда), в котором происходит функционирование и существование агента. У любого агента задано собственное и неизменное *имя*, по которому он может быть идентифицирован. Агенты обладают способностью делать собственные копии с целью последующей их рассылки по разным серверам для совершения необходимой работы. Когда агент отправляется на другой сервер, то по прибытии все имеющиеся у него данные и его код копируются в новый контекст и уничтожаются в пункте, из которого произошёл перенос агента. После завершения действия агента в контексте, он может отправить себя в следующий контекст или вернуть обновившиеся данные в предыдущий контекст, из которого его переслали. Агенты могут прекращать работу самостоятельно или по команде сервера, который переносит их после выключения из контекста в место для хранения.

На Рис.2 схематично изображено устройство агента. У каждого агента имеются входы, то есть внутренние характеристики агента и сведения о положении среды. Так же определены выходы: показатели, влияющие на среду и уведомляющие пользователя о состоянии среды и принятых решениях. Входы преобразуются в выходы посредством решателя, который представляет собой процедуру принятия решений.



Рис. 2: Укрупнённая структура агента [12]

1.3. Многоагентная система

Как было сказано в прошлой части, в многоагентной системе функционирует не меньше двух агентов. Можно описать способ взаимодействия между агентами через понятие "кооперация". Это понятие для наглядности можно изобразить в виде формулы: *кооперация = сотрудничество + координация движений + разрешение конфликтов*.



Рис. 3: Архитектура ядра МАС [12]

Главной частью архитектуры МАС является предметно-независимое ядро (Рис.3), которое состоит из следующих компонентов [12]:

- служба прямого доступа, которая предоставляет доступ к атрибутам агентов;
- служба сообщений, отвечающая за обмен сообщениями между агентами или между агентами и дополнительными системами ядра;
- библиотека классов агентов, обладающая сведениями о классификации агентов в данной МАС;
- сообщество агентов представляющее собой серверную часть, содержащую агентов. Оно также предоставляет функции по загрузке, записи агентов и их свойств, оптимизации работы агентов с ресурсами;
- онтология, то есть предметная база знаний, содержащая определённую информацию об объектах и среде функционирования, сформированную в виде семантической сети.

1.4. Современные международные стандарты создания агентов и платформы МАС

Наиболее известные международные подходы для создания МАС [12]:

- OMG MASIF, созданный Object Management Group, деятельность которого базируется на понятии мобильного агента;
- спецификации FIPA (Foundations for Intelligent Physical Agents), которые действуют исходя из предположения интеллектуальности агента;
- стандарты, разработанные исследовательским подразделением Пентагона – Агентством Передовых Оборонных Научных Исследований (Defense Advanced Research Projects Agency – DARPA), в частности Control of Agent Based Systems.

Спецификации FIPA ориентируются на обеспечение возможности взаимодействия интеллектуальных агентов через стандартизированную коммуникацию агентов и языки контента. Наряду с общими основами коммуникации FIPA специализируется также на протоколах онтологии и переговоров для поддержки взаимодействия в конкретных прикладных сферах (транспортная поддержка, производство, мультимедиа, поддержка сетевого взаимодействия).

Стандарт OMG MASIF нацелен на создание условий для миграции мобильных агентов между мультиагентными системами посредством стандартизированных интерфейсов CORBA IDL.

Организация DARPA инициировала работу по распределению знаний (Knowledge Sharing Effort), в результате которой языки программирования агентов были разделены на синтаксис (syntax), семантику (semantics) и прагматику (pragmatics):

- KIF – Knowledge Interchange Format (syntax);
- Ontolingua – a language for defining sharable ontologies (semantics);

- KQML (Knowledge Query and Manipulation Language) – a high-level interaction language (pragmatics).

Агентные платформы представляют собой один из способов построения распределенных систем и позволяют описать и предоставить доступ всех приложений, работающих на агентной платформе, к необходимым им сервисам. Кроме того, в функции агентной платформы входит распределение агентов, аудит их функционирования и управление стандартными действиями агентов [4].

Java-ориентированные платформы [12]:

- JADE (Java Agent Development Framework) (пример работы с платформой рассмотрен в [9]);
- JACK Intelligent Agents;
- MadKIT;
- AgentBuilder;
- MASON;
- Cougaar (Cognitive Agent Architecture).

Некоторые другие платформы:

- NetLogo;
- VisualBots;
- REPAST;
- CogniTAO.

1.5. Выводы по первой главе

В первой части работы были рассмотрены общие теоретические сведения о мультиагентных системах, что такое интеллектуальный агент

и его структура, как выглядит архитектура ядра многоагентной системы, рассмотрен подход «Агентов и Миров» для разработки МАС и как при таком подходе агенты принимают решения для достижения поставленных целей, перечислены популярные средства, использующиеся при разработке МАС.

Использование мультиагентного подхода в системах управления - перспективная и широко развивающаяся область. Уже существуют различные системы для управления группами интеллектуальных роботов, безопасностью автомобиля, распределения заказов такси (например, Addison Lee) и т.д. Применение МАС позволяет поднять эффективность использования ресурсов до 20-40% по сравнению с использованием традиционных систем, работающих на основе детерминированных алгоритмов.

2. Использование мультиагентного подхода для решения задач планирования

2.1. Некоторые сведения из логистики

«*Логистика* – это учение о планировании, управлении и контроле движения материальных, информационных и финансовых ресурсов в различных системах». ¹

Процесс перемещения материальных объектов от первичного источника сырья до потребителя происходит с помощью транспортов различных видов. В некоторых случаях стоимость транспортировки может составлять до 50% от стоимости общих средств, потраченных на логистику [10].

Транспортная логистика – это один из разделов логистики, занимающийся задачами процесса организации и транспортировки груза из начального пункта в конечный по оптимальному маршруту. В конечном итоге после использования транспортной логистической системы для предприятия должно выполняться «шесть правил логистики» — требуемый груз, в указанном месте, в заданное время, в необходимом количестве, определённого качества, с минимальными затратами.

Транспорт представляют как систему, состоящую из трёх компонент [8]:

- Транспорт общего пользования включающий такие типы транспортов, как железнодорожный, водный (морской и речной), автомобильный, воздушный, трубопроводный.
- Транспорт необщего пользования либо являющийся внутрипроизводственным, либо принадлежащий нетранспортным предприятиям (как правило, является составной частью производственных систем).

¹Определение было сформулировано и принято Первым европейским конгрессом по логистике, проходившем в Берлине с 20 по 22 марта 1974 года

- Транспорт специализированного назначения - предназначен для доставки спецгрузов различного назначения.

В нашей работе для создания МАС рассматривается транспорт общего пользования: грузовики, совершающие междугородные перевозки. Целью данной части является формальное описание процесса распределения заказов и планирование маршрутов для их последующей транспортировки.

Рассматривается сложная задача *планирования, распределения и оптимизации перевозки*, когда состав ресурсов и заказов может меняться «на лету» [13]: поломка грузовика в пути и необходимость решения данной проблемы в зависимости от сложности поломки; болезнь водителя, которая приводит к перепланированию расписания перевозок; появление новых заказов и необходимость их обработки и своевременного выполнения.

Современные производственные системы перестают быть централизованными и становятся распределёнными [4]. Вместо предварительного расчета и составления единого расписания грузоперевозок требуется учет влияния внешних факторов в режиме реального времени. Мультиагентные технологии не только обеспечивают эти требования, но и позволяют решать сложные задачи с большим числом переменных путём разбиения на более простые задачи.

2.2. Постановка задачи адаптивного планирования

Опишем формально задачу для некоторой организации, занимающейся междугородными грузоперевозками:

1. Имеется парк грузовиков, состоящий из N , оборудованных GPS датчиками, положение которых отражается на карте.
2. В режиме реального времени диспетчерская служба принимает новые заказы на междугородные перевозки или сообщения о каких-либо незапланированных событиях (например: задержки, поломки и т.д.), которые необходимо обрабатывать и генерировать план

реализации новых перевозок или устранение возникших неполадок, учитывая текущие ресурсы [14].

3. Постоянно случающиеся события незамедлительно приводят к изменениям, которые должны учитываться и вноситься в план использования ресурсов без остановок и перезапуска системы, путем адаптивного изменения расписания с использованием как свободных окон времени, так и подвижками во времени и переброской на другие ресурсы ранее распределённых заказов.



Рис. 4: Полный цикл автономного управления ресурсами [11]

Рассмотрим полный цикл автономного управления ресурсами (Рис.4):

- система в индивидуальном порядке реагирует на каждое событие;
- в результате процесса начального построения расписания или его последующего перепланирования ввиду появившихся изменений получаем динамическое планирование;
- если по окончании построения расписания остаётся время, то возможна оптимизация полученного плана;

- необходим постоянный мониторинг и контроль исполнения планов в условиях самостоятельного контроля системой исполнения разрабатываемого ею же плана;
- в случае обнаружения расхождения между планом и фактом будет совершено перепланирование расписания;
- постоянное обучение из уже накопленного опыта.

2.3. Концепция сетей потребностей и возможностей (ПВ-сеть)

Мультиагентный подход позволяет создавать программных агентов, которые стараются оптимизировать свои целевые параметры [6]: заказ грузовой компании должен быть выполнен точно в срок и с минимальными затратами, ресурс должен использоваться как можно более эффективно и не простаивать попусту. Изначально агенты совершают работу автономно, ни с кем не коммуницируя, в следствие чего, очень быстро, если ресурсы свободны; но, при возникновении конфликтов интересов между агентами, способны вступать в переговоры, делать уступки и добиваться согласованного решения в интересах объединяющего их целого (например, водителя или автопарка).

Будем рассматривать задачу адаптивного планирования [16] в рамках новой концепции, речь о которой пойдёт дальше, для того, чтобы переформулировать задачу построения расписания в условиях задачи, описанной математическим языком.

При использовании данной концепции каждой заявке, заказу и другим потребностям и возможностям (оборудование, персонал, транспорт и т.д.) будем сопоставлять своего программного агента (рис.5), который будет общаться с другими агентами, стараясь максимально выполнить свои желания (например, максимизировать доход от перевозки, перевозить груз в машине, оборудованной холодильником и т.п.). Агент обладает возможностью договариваться с другими агентами для планирования исполнения заказа «точно-в-срок» или «как можно раньше».

Коммуникация агентов позволяет обеспечить поддержку коллективного согласования и принятия решений в режиме реального времени на различных этапах планирования и исполнения производственного плана в различных подразделениях, работающих совместно над решением общих задач [3].

Между конкурирующими и кооперирующимися ПВ-агентами постоянно происходит поиск соответствий, приводящий к решению любой сложной задачи в виде динамической сети связей, гибко изменяемую в реальном времени.

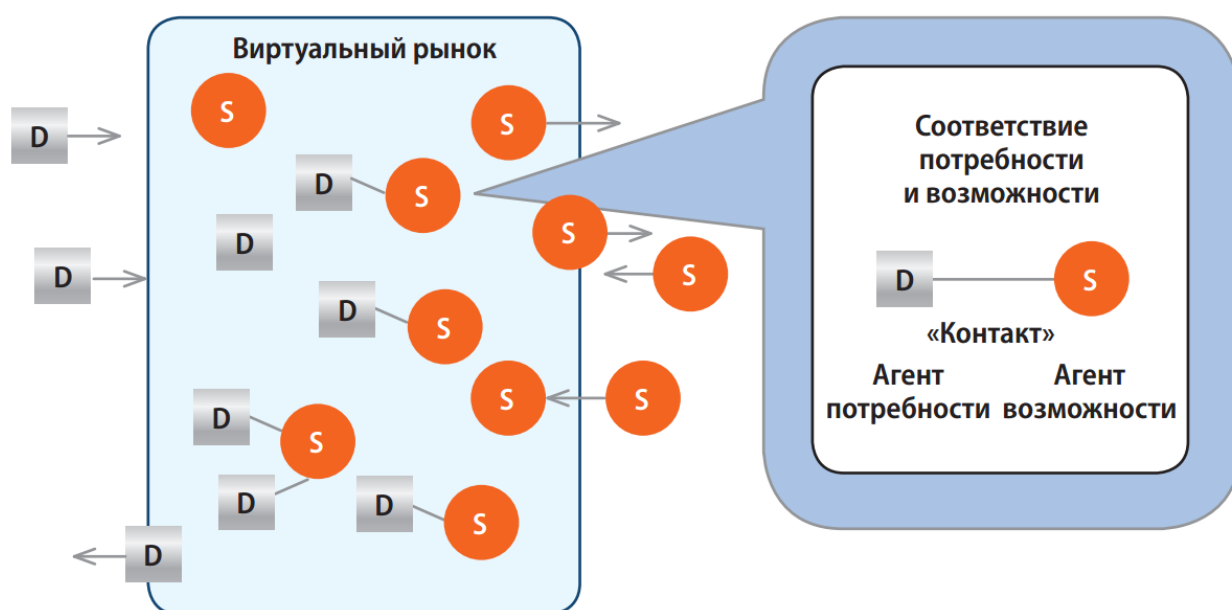


Рис. 5: Виртуальный рынок ПВ-сети для построения расписаний [11]

Перейдём к формальному рассмотрению задачи построения расписания, в основе которой лежит поиск консенсуса между агентами на виртуальном рынке в ПВ-сетях.

Пусть у всех агентов спроса и предложений заданы собственные цели, ограничения, критерии: срок, стоимость, приоритет и т.п. Будем рассматривать важность (обладающую возможностью изменяться по ходу формирования или перестроения расписания) каждого показателя в виде весовых коэффициентов в линейной комбинации заданных критериев для определённого момента планирования.

Далее для каждого агента, который характеризуется некоторыми определёнными показателями, можно рассмотреть функцию удовлетво-

рённости, смысл которой заключается в нахождении отклонения текущего значения функции от данного идеального значения по любому из критериев на данном шаге нахождения расписания для фиксированного агента. На Рис.6 для наглядности приведён пример функции удовлетворённости.

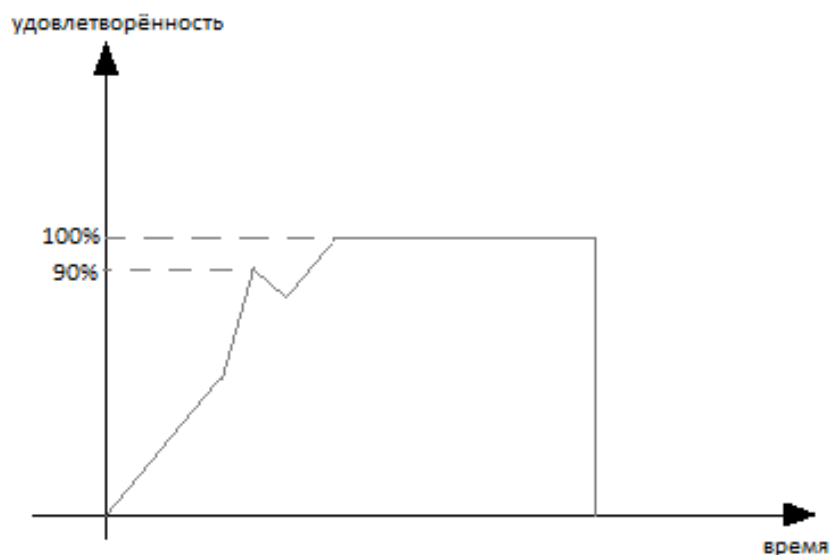


Рис. 6: Пример функции удовлетворённости

С другой стороны, деятельность агентов так же зависит от функции бонусов/штрафов. На Рис.7 приведён пример такой функции.

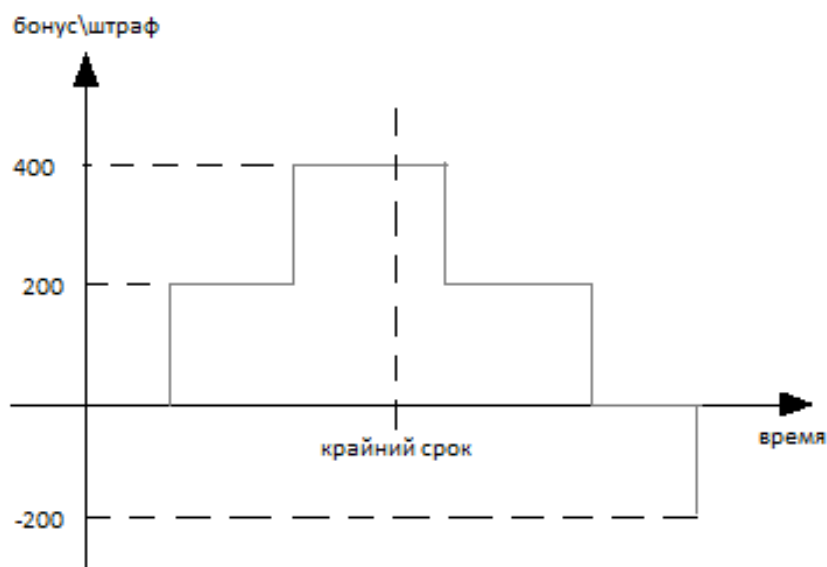


Рис. 7: Пример функции бонуса/штрафа

Пусть для потребности j определено несколько индивидуальных критериев x_i и имеются значения идеальных показателей x_{ij}^{id} . Для каждого агента потребности j вводится нормализованная функция удовлетворённости, которая считается по компоненте i , например, как кусочно-линейная функция $f_{ij}^{need}(x_i - x_{ij}^{id})$. График данной функции будет иметь своё максимальное значение в точке с заданным идеальным значением критерия. Теперь для каждого рассматриваемого агента и для каждого критерия i с определённым весовым коэффициентом a_{ij}^{need} считаем значение функции удовлетворенности.

На самом деле, целевая функция каждого агента может быть переформулирована в виде задачи максимизации удовлетворенности y_j^{need} потребности j (индекс *need* означает, что значение принадлежит агентам потребности):

$$y_j^{need} = \sum_i a_{ij}^{need} * f_{ij}^{need}(x_i - x_{ij}^{id}), \quad \sum_i a_{ij}^{need} = 1 \quad (1)$$

По аналогии можно формально описать задачу поиска состояний x_{ij}^* агентов потребности j , которые максимизируют значение всех заказов:

$$y^{need} = \sum_j b_j^{need} y_j^{need} = \sum_j b_j^{need} \sum_i a_{ij}^{need} * f_{ij}^{need}(x_i - x_{ij}^{id}) \quad (2)$$

$$y^{need*} = \max_{x_i} y^{need} \quad (3)$$

b_j^{need} -вес потребности, который в ходе изменения показывает важность критерия в тот или иной момент.

По аналогии и для агента ресурса по критерию z_k можно задать функцию удовлетворённости $f_{ij}^{res}(z_i - z_{ij}^{id})$ с весом a_{kl}^{res} критерия k для ресурса l и значением ресурса b_i^{res} для системы:

$$y^{res} = \sum_l b_l^{res} y_j^{res} = \sum_l b_l^{res} \sum_k a_{kl}^{res} * f_{kl}^{res}(z_k - z_{kl}^{id}) \quad (4)$$

$$y^{res*} = \max_{z_k} y^{res} \quad (5)$$

Переменные x и z принадлежат области критериев заказов и ресур-

сов, L и K – размерности пространств, индекс res показывает принадлежность к агентам ресурсов.

Теперь задача построения расписания переформулирована в условиях задачи максимизации удовлетворённости.

2.4. Модель организации перевозок

Каждый грузовик имеет определенный объём, который он способен вместить в себя, поэтому будет естественно задать ограничение сверху на возможный объём перевозимого груза [5]:

$$x_i(t) \leq V_i, \quad (6)$$

где $x_i(t)$ -задействованный в момент времени t , V_i -максимальный объём i -ого грузовика. Общая прибыль системы высчитывается как сумма прибылей всех агентов грузовиков:

$$P = \sum_i P_i \quad (7)$$

N_i -набор заказов, выполненных i -ым грузовиком. Тогда прибыль отдельного агента рассчитывается по формуле:

$$P_i = \sum_{j \in N_i} [(c_j - q_i)t_{ij} - q_i t'_{ij}] \quad (8)$$

Расчеты ведутся для грузовика i по всем выполненным заказам j , где c_j - стоимость выполнения заказа j в единицу времени, q_i - стоимость переезда грузовика в единицу времени t_{ij} - время выполнения заказа j , t'_{ij} - время переезда грузовика до пункта расположения заказа j .

Модель 1 («с возвращением на базу»). Доставка груза выполняется на грузовике, который был свободен в момент поступления заказа. Если груз находится в другом пункте (то есть погрузка должна совершиться в пункте, отличном от того, где находится грузовик), то машина должна совершить переезд в другой пункт до момента начала погрузки. При данной модели бронирование грузовика не может быть отменено (на-

пример, в случае появления более выгодного заказа для перевозки на данной машине). По окончании грузоперевозки машина должна вернуться в пункт базирования.

Модель 2 («без возвращения на базу»). Данная модель аналогична предыдущей с тем отличием, что после доставки заказа в конечный пункт грузовик не возвращается на базу, а остаётся в пункте назначения, где ждёт следующий заказ.

Модель 3 («опоздание со штрафами»). Разрешается начинать исполнение заказа позже того времени, что установил заказчик. Однако из прибыли, полученной за выполнение заказа с опозданием, вычитается штраф, соответствующий времени отсрочки. Если штраф превышает возможную прибыль, то такой заказ нецелесообразно исполнять, поэтому он не будет планироваться. После исполнения заказа машина остается в пункте назначения, аналогично модели 2 и ожидает там следующий заказ. Для расчета прибыли машины с учетом штрафов используется формула:

$$P_i(t) = \sum_{j \in T_i} [(c_j - q_i)t_{ij} - q_i t'_{ij}] + \sum_{k \in L_i} [(c_k - q_i)t_{ik} - q_i t'_{ik} - p_k t''_{ik}] \quad (9)$$

где T_i - множество заказов грузовика i , выполненных без опоздания, L_i - множество заказов, выполненных грузовиком i с опозданием, t''_{ik} - время опоздания исполнения k -ого заказа грузовиком i , p_k - штраф на единицу времени опоздания.

Модель 4 («адаптивное планирование со штрафами»). Модель аналогична предыдущей с тем отличием, что допускается бронирование грузовика для нового заказа при условии, что прибыль от нового заказа будет превышать возможную прибыль от уже запланированного заказа. При поступлении нового заказа произойдёт перераспределение уже запланированных заказов и найдётся новое, более выгодное планирование.

2.5. Процесс планирования

Опишем поведение системы планирования в виде бизнес-процесса с использованием Visio. Алгоритм поиска решений в случае поступления нового заказа может быть следующим:

1. в момент приёма нового заказа запускается система планирования расписания;
2. начинается логическая обработка события, которая ищет наилучшее решение для перевозки заказа с учётом имеющихся ресурсов, имеющегося опыта;
3. согласно алгоритму работы системы подбирается наиболее подходящее решение;
4. если в случае поиска решения, выясняется, что невозможно рационально спланировать перевозку, то клиент информируется об этом;
5. в конце планирования в журнал заносятся результаты работы системы с проставлением даты.

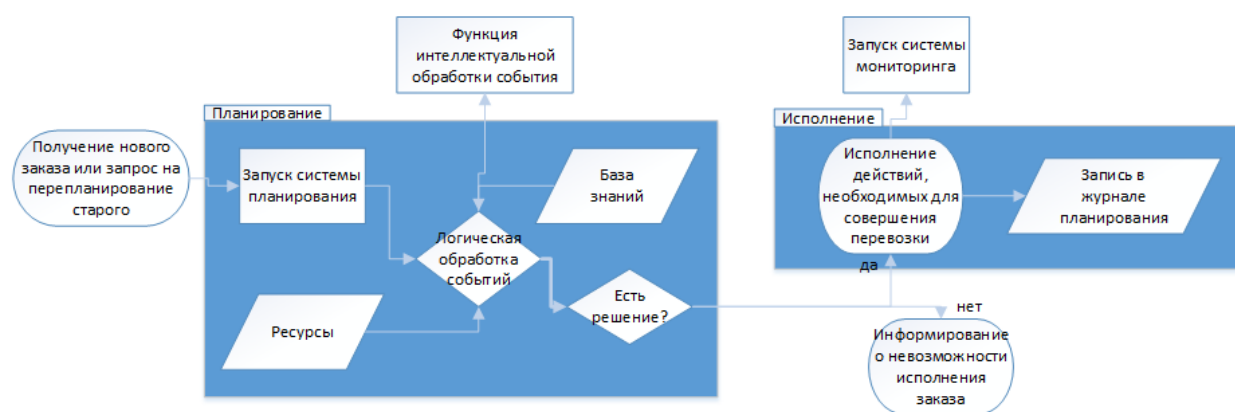


Рис. 8: Схема процесса планирования расписания с применением мультиагентного подхода

2.6. Модельный пример расчета прибыли

Рассмотрим модельный пример расчета прибыли для модели 3 для одного грузовика.

Имеются 4 пункта с заданными характеристиками, описывающими количество дней, необходимое для переезда из одного пункта в другой (Рис.9).

—	Пункт 1	Пункт 2	Пункт 3	Пункт 4
Пункт 1	0	1	1	2
Пункт 2	1	0	2	1
Пункт 3	1	2	0	1
Пункт 4	2	1	1	0

Рис. 9: Расстояние между пунктами [1]

Изначально грузовик располагается в пункте 1.

В разные моменты времени поступают заказы 1-5 на грузоперевозки (Рис.10). Для удобства будем использовать ограничение сверху на дни, в которые можно совершить перевозку: зададим горизонт планирования $t = 10$. Стоимость выполнения заказа постоянна и равна $c = 3$ у.е./день. Каждый день простоя грузовика характеризуется штрафом в размере $q_a = 0,3$ у.е. Каждый день работы грузовика, будь то исполнение заказа или переезд из одного пункта в другой, приводит к постоянным затратам, равным $q = 1$ у.е./день. При выполнении заказа с опозданием за каждый просроченный день налагается штраф в размере $pp=0,6$ у.е.

Расчет прибыли происходит по мере планирования заказов:

1. Для выполнения заказа 1 грузовик должен выехать в момент времени $t = 1$ из пункта 1 в пункт 4. Путь переезда пустого грузовика займёт 2 дня, поэтому на момент времени $t = 3$ прибыль грузовика будет равняться $P = -q * 2 = -2$.
2. Непосредственное выполнение заказа 1 займёт 2 дня. К моменту времени $t = 5$ грузовик перевезёт заказ 1 из пункта 4 в пункт 1.

Характеристика	Номер заказа				
	1	2	3	4	5
Время поступления	1	3	5	6	7
Время начала	3	4	7	8	9
Время окончания	5	5	9	9	10
Откуда	4	3	1	4	3
Куда	1	1	4	3	1

Рис. 10: Поступившие заказы [1]

Прибыль будет составлять $P = -2 + (c - q) * 2 = -2 + 2 * 2 = 2$.

3. На данном этапе $t = 5$ грузовик находится в пункте 1 с прибылью $P = 2$, тем временем в момент времени $t = 3$ поступил заказ 2 и на этапе $t = 5$ поступил заказ 3. Агенту грузовика необходимо определить, что будет выгоднее: исполнить заказ 2 с опозданием или не исполнять его совсем, а через 2 дня исполнить заказ 3 из пункта 1.
4. Рассмотрим исходы обоих вариантов. Для выполнения заказа 2 необходимо переехать из пункта 1 в пункт 3, что займёт 1 день. В случае исполнения этого заказа общая прибыль составит $P = 2 - pr * 2 - 1 * q + (c - q) * 1 = 2 - 1,2 - 1 + 2 = 1,8$ и на момент времени $t = 7$ грузовик вновь окажется в пункте 1. Кажется, что это не выгодно, так как общая прибыль грузовика уменьшилась, однако стоит заметить, что в случае отказа от заказа 2, грузовик в ожидании заказа 3 вынужден будет простаивать 2 дня, что ведёт к потерям за счет простоя и на момент времени $t = 7$ общая прибыль грузовика составит $P = 2 - q_a * 2 = 2 - 0,6 = 1,4$.
5. Агенту грузовика оказалось выгодно выполнить с опозданием заказ 2, поэтому машина совершит данную грузоперевозку с задержкой и выплатой соответствующих штрафов, затем исполнит заказ 3, выполнение которого займёт 2 дня. На момент $t = 9$ грузовик окажется в пункте 4 с прибылью $P = 1,8 + (c - q) * 2 = 1,8 + 2 * 2 = 5,8$.

6. К моменту $t = 9$ есть новый заказ 5 в пункте 3 с моментом начала выполнения $t = 9$, однако переезд из текущего пункта в пункт заказа займёт 1 день, что сдвинет заказ за горизонт $t = 10$. Поэтому агент грузовика от него отказывается. Однако есть просроченный заказ 4.
7. Выполнение заказа 4 не требует переезд. Прирост прибыли при его выполнении составит $dP = (c - q) * 1 - pp * 1 = 2 - 0,6 = 1,4$. Однако при отказе от этого заказа грузовик вынужден будет простаивать 1 день, что приводит к отрицательному приросту прибыли $dP = -0,3$ в силу бездействия машины. Поэтому агенту грузовика выгодно исполнить заказ 4 с сопутствующими штрафами.

Итого: заказы 1,3 выполняются без опоздания, заказ 2 с задержкой на 2 дня, заказ 4 с опозданием на день. Заказ 5 не выполняется на данной машине. Общая прибыль грузовика за 10 дней составит $P = 5,8 + 1,4 = 7,2$. На Рис.11 для наглядности показано изменение общей прибыли грузовика в зависимости от времени.

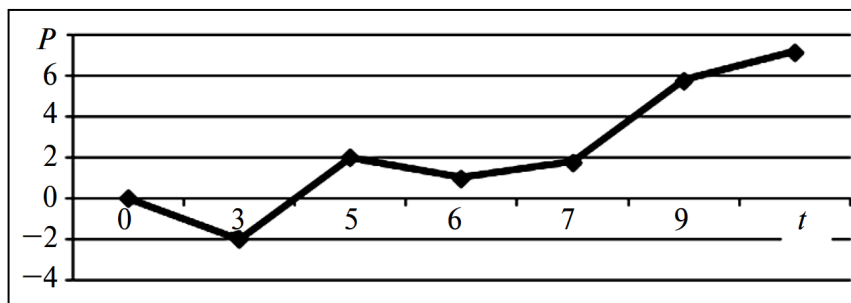


Рис. 11: Прибыль агента грузовика в зависимости от времени [1]

2.7. Выводы по второй главе

Во второй главе были рассмотрены базовые сведения для задачи транспортной логистики, в отношении которой можно применять мультиагентный подход. Цель транспортной логистики не только в перевозке грузов, но также и в предварительном планировании перевозок: в со-

ставлении эффективных маршрутов и расписаний. В частности в этой части работы:

- формально описали задачу адаптивного планирования;
- описали возможность применения мультиагентных технологий для задачи планирования;
- рассмотрели различные модели организации грузоперевозок;
- построили пример расчета прибыли для одной из моделей организации перевозок.

Целью следующей главы является разработка собственной МАС для управления грузовиком, имеющим маршрут и расписание, составленные на основе мультиагентного подхода, описанного во второй главе.

3. Разработка мультиагентной системы для задачи грузоперевозки

3.1. Общее описание системы

Перед нами стоит задача перевозок уже распланированных заказов. Имеется парк, состоящий из n грузовиков, для каждого из которых определено расписание и маршруты движения, построенные согласно принципам, изложенным во второй главе. Формально опишем систему отслеживания передвижения машины и корректировки её расписания в связи с незапланированными событиями (пробки, закрытие дороги, снегопад и т.д.).

Каждая машина оснащена средствами GPS-навигации. У каждой машины есть пять агентов, отслеживающих определённые показатели:

- погоду вдоль трассы;
- соблюдение и корректировку графика движения (оптимальную скорость);
- время погрузки/разгрузки;
- самочувствие водителя;
- техническое состояние автомобиля.

Для машины определён идеальный (то есть запланированный диспетчерским центром) маршрут, у которого заданы начальный, конечный и промежуточные пункты погрузки/разгрузки (пусть они условно попадают в квадрат 100×100 [5], то есть $(x, y) \in [0, 100] \times [0, 100]$) и задано время, в которое машина должна приезжать/уезжать из этих пунктов.

На Рис.12 показан пример: в момент времени T_1 грузовик выдвигается из пункта A в пункт B , время прибытия в пункт B в идеале равно T_2 , затем в интервале с T_2 до T_2' грузовик стоит на погрузке/разгрузке. В момент времени T_2' машина выезжает из пункта B и приезжает в

Таблица 1: координаты начала и конца маршрута

Начальный пункт	Конечный пункт
(10, 43)	(1, 19)
(34, 55)	(16, 9)
(14, 10)	(21, 90)
(73, 17)	(10, 30)
(19, 40)	(39, 29)
(59, 59)	(68, 30)
(49, 3)	(49, 9)
(93, 57)	(2, 59)
(28, 96)	(38, 69)
(38, 69)	(29, 60)
(48, 60)	(60, 40)

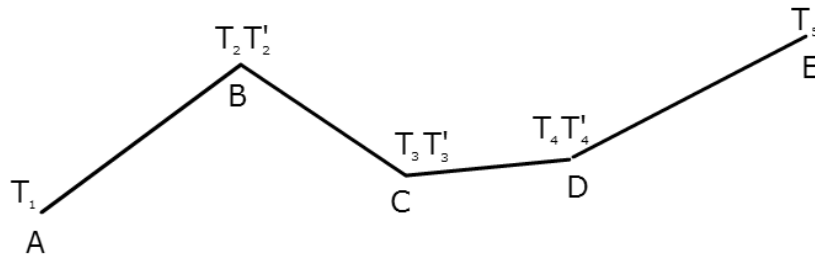


Рис. 12: Пример маршрута грузовика

пункт C ко времени T_3 . Далее стоит на погрузке/разгрузке в пункте C , затем аналогично для пункта D . Грузовик завершает свой маршрут в пункте E в момент времени T_5 .

Таким образом для каждого отрезка пути ($A \rightarrow B$, $B \rightarrow C$ и т.д.) имеем заданную идеальную постоянную скорость для участка пути:

$$v_{ij} = \frac{\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}}{T_j - T_i} \quad (10)$$

где величины с индексом i соответствуют координатам и времени для пункта отправки (начального или промежуточных), величины с индексом j – для пункта прибытия (конечного или промежуточных).

В итоге имеем для каждой машины, выполняющей перевозку, за-

планированный маршрут (время погрузок/разгрузок, путь из пункта в пункт), и реальное положение грузовика, информация о котором поступает с помощью GPS-системы.

Перейдём к более конкретному рассмотрению агентов разрабатываемой системы. Дадим название агенту по выполняемой им функции.

3.2. Агент «Погода вдоль трассы»

Агент считывает информацию о температуре и осадках вдоль маршрута грузовика. У агента задана информация о допустимом отклонении температуры и допустимых уровнях осадков. Если показатели в рамках допустимых, то агент «спит», а если какой-либо показатель выходит за рамки, то агент включает обогрев, подкорректировать скорость движения, отправляет в диспетчерский центр запрос о перепланировании маршрута и т.д. Несколько примеров:

1. В осенний период перевозят конфеты. Перевозку нельзя осуществлять при температуре ниже 5 градусов. Когда машина выезжала с товаром, то температура была плюсовая, но в одном из ожидаемых мест остановки температура упала до минус 10. Тогда агент погоды либо информирует водителя о необходимости обогрева кузова на участке дороги с низкой температурой, либо самостоятельно включает обогрев в нужное время.
2. В зимний период на большом участке дороги начался сильный снегопад. Уровень осадков стал недопустим для движения с ожидаемой скоростью. Агент погоды отправляет сообщение в диспетчерский центр, который, в свою очередь, либо выберет другой маршрут, либо скорректирует скорость движения (например, на заснеженном участке скорость будет снижена, а затем надо будет увеличить скорость, чтобы нагнать расписание).

Алгоритм работы агента:

procedure *WeatherHandling*

t = текущее время, (x, y) = текущие координаты;
 задание сдвига s для времени t ;
 максимально возможная температура t_{max} ;
 минимально возможная температура t_{min} ;
 максимально возможный уровень осадков w ;
repeat
 if $GetTemperature(t, x, y) > t_{max}$ **then**
 включить охлаждение;
 end if
 if $GetTemperature(t, x, y) < t_{min}$ **then**
 включить обогрев;
 end if
 if $GetPrecipitation(t, x, y) > w$ **then**
 отправить диспетчеру запрос о перепланировании маршрута;
 end if
 $t = t + s$;
 (x, y) = ожидаемые координаты в момент времени t ;
 until достигнуто ожидаемое время прибытия в пункт остановки;
end procedure
function $GetTemperature(t, x, y)$
 получение температуры в точке (x, y) в момент времени t ;
end function
function $GetPrecipitation(t, x, y)$
 получение уровня осадков в точке (x, y) в момент времени t ;
end function

Данный агент начинает свою работу в момент оправки из пункта погрузки/разгрузки, либо при перепланировании маршрута движения.

Агенту соответствует следующая таблица из базы данных:

В таблицу заносятся погодные условия в определённый момент времени и локации для i -ого заказа.

weather_ID	order_ID	temperature	rainfall	location	date	time
1	1	+1	0 мм	(10, 20)	2015-10-22	10:20:00
2	3	-10	0,01 мм	(5, 76)	2016-12-09	15:00:00
3	3	-13	10 мм	(35, 24)	2016-02-09	10:00:00

Рис. 13: WeatherHandling

3.3. Агент «Оптимальная скорость»

Данный агент сравнивает ожидаемые координаты, время, скорость с действительными значениями, полученными с помощью GPS-системы. У агента заданы координаты всех пунктов загрузки/разгрузки, время отправки и прибытия в эти пункты.

Текущая средняя скорость рассчитывается по формуле:

$$v_{ij} = \frac{\sqrt{(x_i - x)^2 + (y_i - y)^2}}{T - T_i} \quad (11)$$

где величины с индексом i соответствуют координатам пункта отправки и времени выезда из пункта отправки.

Агенту задана некоторая величина, характеризующая максимально допустимую величину разброса ε текущей средней скорости от заданной средней скорости. Если v_i будет меньше $v_{ij} - \varepsilon$, то агент посылает сообщение в диспетчерский центр о задержке. В свою очередь диспетчерский центр может послать водителю сообщение о необходимости увеличить скорость, отправить новое расписание и т.д. И наоборот, если v_i будет больше $v_{ij} + \varepsilon$, то водителю посылается сообщение о необходимости снизить скорость, потому что заблаговременный приезд, например, мешает работе на разгрузочном пункте.

Алгоритм работы агента:

procedure *SpeedControl*

 задание сдвига для времени s ;

t = текущее время, (x, y) = текущие координаты;

repeat

if *GetSpeed*(t, x, y) < $v_{ij} - \varepsilon$ **then**

 отправить в диспетчерский центр сообщение о задержке;

```

end if
if  $GetSpeed(t, x, y) > v_{ij} + \varepsilon$  then
    отправить водителю запрос снизить скорость;
end if
 $t = t + s$ ;
задержка на время  $s$ ;
 $(x, y)$  = ожидаемые координаты в момент времени  $t$ ;
until не достигнут пункт погрузки/разгрузки;
end procedure
function  $GetSpeed(t, x, y)$ 
    возвращает значение, полученное по формуле (11);
end function

```

speed_ID	order_ID	speed	location	date	time
1	2	67	(39, 89)	2015-06-09	12:40:00
2	3	58	(10, 20)	2015-09-15	08:35:08

Рис. 14: SpeedControl

В таблицу заносятся показатели скорости для грузовика, везущего i -ый заказ в момент времени $t = \text{текущее время}$ в точке (x, y) , координаты которой определяются с помощью GPS-системы.

3.4. Агент «Погрузка/разгрузка»

Этот агент отслеживает время погрузки/разгрузки. В момент прибытия в i -ую зону остановки "просыпается" агент погрузки/разгрузки и начинает свою работу.

Алгоритм работы агента:

```

procedure  $ParkingTime$ 
     $t$  = текущее время;
     $s = false$ ;
    repeat
        if  $t > T_i$  then

```

```

         $s = true;$ 
    end if
    задержка на  $t = 1$ ;
     $t = t + 1$ ;
until закончена погрузка/разгрузка;
if  $s = true$  then
    отправить диспетчеру сообщение о задержке погрузки/разгрузки;
    посчитать необходимую скорость  $v$  движения,
    чтобы приехать в следующий пункт по расписанию;
    if  $(v > v_{ij} + \varepsilon)$  then
        отправить диспетчеру сообщение о задержке
        прибытия в следующий пункт;
    else
        информировать водителя повысить скорость;
    end if
end if
if  $t < T_i'$  then
    отправить сообщение диспетчеру об окончании
    погрузки/разгрузки раньше ожидаемого времени;
end if
end procedure

```

parking_ID	order_ID	start_time	finish_time	date
1	1	09:00:00	10:01:44	2015-10-07
2	3	16:14:00	16:59:16	2016-01-06

Рис. 15: ParkingTime

В таблицу заносится время начала и окончания погрузки/разгрузки для грузовика, выполняющего i -ый заказ.

3.5. Агент «Самочувствие водителя»

Работа данного агента связана с наличием у водителя ручного браслета, считывающего показатели состояния здоровья водителя (сердцебиение, температура и т.д.). При отклонении показателей от средне допустимых, агент либо предпринимает действия по отношению к водителю, направленные на улучшение его показателей, либо отправляет сообщение диспетчеру о возникших неприятностях.

Примеры действия агента самочувствия:

- Существуют различные браслеты, не позволяющие водителям заснуть. Некоторые реагируют на наклон головы, другие на движение глаз. В случае засыпания водителя браслет начинает вибрировать.
- Если в случае серьёзной аварии у водителя остановится сердце, то агент посылает сообщение диспетчеру о критическом состоянии водителя .

Алгоритм работы агента:

procedure *DriverCondition*

repeat

if изменение состояния не опасно для жизни **then**
 совершение действий, корректирующих состояние
 или информирование водителя;

end if

if показатели характеристик опасны для жизни **then**
 информировать диспетчера о критическом состоянии;
 end if

until браслет надет;

end procedure

В таблицу заносятся показатели пульса и температуры водителя, который совершает i -ую грузоперевозку, в момент времени t .

condition_ID	order_ID	pulse	temperature	date	time
1	1	63	36,7	2015-12-08	14:23:00
2	3	99	37,1	2015-12-28	09:31:00

Рис. 16: DriverCondition

3.6. Агент «Техническое состояние автомобиля»

Автоматическая диагностика автомобиля - вещь распространённая и широко используемая. Системы диагностики, как правило, разнятся между собой в зависимости от машин, на которых они функционируют. Однако все системы работают по одному принципу: существует некий блок управления, на который поступают считываемые характеристики различных датчиков машины, отвечающие за разнообразные режимы работы процесса эксплуатации автомобиля (например, режим описывающий процесс запуска, холостого хода, торможения, прогрева, разгона и т.д.). Существует два типа показания датчиков: они могут быть статическими (дискретными) или динамическими (изменяющие свои значения с изменением времени). Статический показания датчиков чаще всего характеризуются таким определённым значением, как наличие сигнала или его отсутствие (например, датчики характеризующие режим работы подушек безопасности могут передавать только два показателя: подушки были использованы, либо не были). Динамические показания датчиков, напротив, обычно показывают изменения параметра и, как правило, показатели этих датчиков проверяются на допустимые диапазоны характеристики того или иного режима работы автомобиля (то есть верхний и/или нижний границы показателей, выход за пределы которых считается нарушением работы какого-то модуля). Также вне зависимости от типа диагностической системы запоминаются и хранятся статические данные: «коды ошибок» и динамические характеристики [7].

Рассмотрим теперь принципы реагирования системы диагностики автомобиля на датчики, характеризующиеся статическими показателями и на датчики, определяющиеся динамическими значениями. Как

правило, рассматривая статические показатели датчиков, система самодиагностики будет реагировать в условиях отсутствия электрического контакта (например, будет посылаться сигнал о неисправности датчика). На датчики с динамическими характеристиками система реагирует по-другому: ведётся отслеживание изменения показателей датчиков по данным, заданным в памяти устройства управления. Датчики не обязательно могут характеризоваться только статическими или только динамическими показателями, также возможен вариант, когда определённый датчик будет проверяться как на электрический контакт, так и на не выходение динамического показателя за границы допустимых значений характеристики. В этом случае для одного датчика могут быть определены сразу две ошибки: отсутствие сигнала, либо наличие показателей, выходящие за границы допустимых.

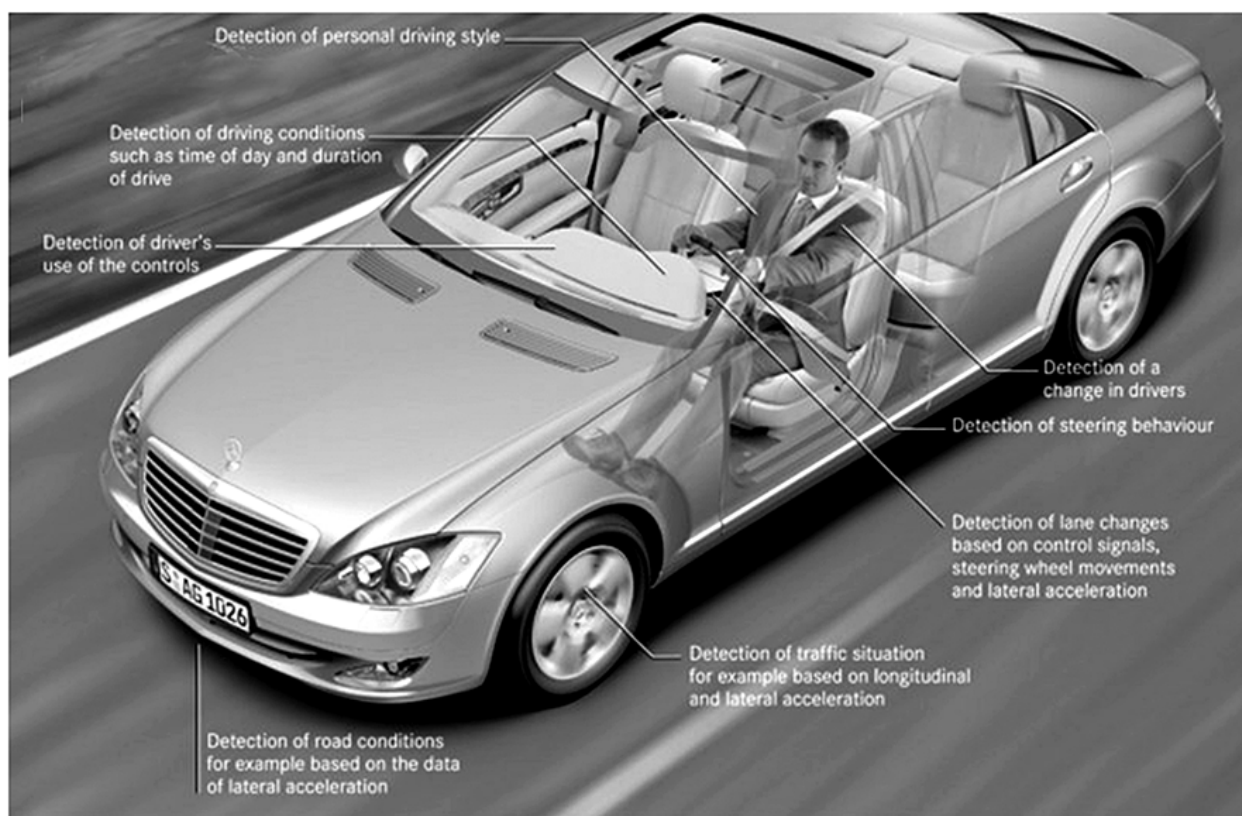


Рис. 17: Схема анализа информации, полученной от датчиков [12]

Устройство управления содержит в себе несколько отдельных блоков: одна часть отвечает за двигатель – ECU (Engine Control Unit) или ECM (Engine Control Module), другой блок соответствует антиблокиро-

вочной системе тормозов – ABS, ещё один блок отвечает за действие подушек безопасности – SRS (Air Bag Supplemental Restraint System), автоматической коробки передач соответствует A/T (Electronic Automatic Transaxles), также существуют и другие блоки. Однако в случае возникновения сигнала об ошибке система диагностики автомобиля обязательно должна ответить унифицировано [15]:

- в начале система должна определить тип неисправности по имеющимся в системе данным (классифицировать неисправность по номеру) и внести эту ошибку в долговременную память;
- затем система должна совершить ряд действий, направленных на исправление ошибки или минимизацию ущерба от неё, согласно сценарию действий, соответствующему данному типу ошибки.

По окончании исполнения действий, необходимых для минимизации ущерба, сохранённые в памяти коды ошибок считываются либо специальным прибором (сканером), либо вручную, при помощи определённой процедуры, которая вводит ECU в режим индикации кодов самодиагностики.

Рассмотрим агента технического состояния машины с позиции нашей многоагентной системы. Агент отвечает за информирование водителя либо диспетчера о возможных или возникших неполадках машины. На самом деле существуют сложные многоагентные системы, следящие за состоянием автомобиля, но в данной системе будем использовать упрощённую версию, состоящую из одного агента состояния машины. У агента есть некий набор характеристик автомобиля (x_1, x_2, \dots, x_n) (например, величина свободного хода педали тормоза, развал схождения и т. д.), которые он отслеживает и величины допустимых разбросов этих характеристик. Также задан вектор идеальных значений.

Агент сравнивает текущие показатели характеристик автомобиля с вектором идеальных значений. Если тот или иной показатель выходит за рамки допустимого, то агент отправляет сообщение диспетчеру, либо рекомендацию водителю о возможности исправления неполадки.

Алгоритм работы агента:

procedure *LorryCondition*

t = время отъезда из пункта загрузки/разгрузки;

задание задержки s времени t ;

repeat

GetCharacteristic(t);

for $i = 0$; $i < \text{size}(\text{CharacteristicList})$; $i++$ **do**

if *CharacteristicList*[i] не допустим;

then информировать водителя и диспетчерский центр

о критическом состоянии машины и

невозможности дальнейшего движения;

end if

end for

$t = t + s$;

задержка на время s ;

until не достигнут пункт остановки;

end procedure

procedure *GetCharacteristic*(t)

for $i = 0$; $i < \text{size}(\text{CharacteristicList})$; $i++$ **do**

CharacteristicList[i] = *RequestCharacteristic*(i, t);

end for

end procedure

condition_ID	order_ID	airbag	date	time
1	1	not used	2015-10-06	09:10:00
2	3	not used	2015-12-31	14:19:00

Рис. 18: LorryCondition

В таблицу заносятся показатели технического состояния машины, совершающей i -ую перевозку в момент времени t (в данном примере в таблицу заносится состояние подушки безопасности, показывающее, была ли она использована).

3.7. Процесс мониторинга и архитектура уровня приложений и инфраструктуры

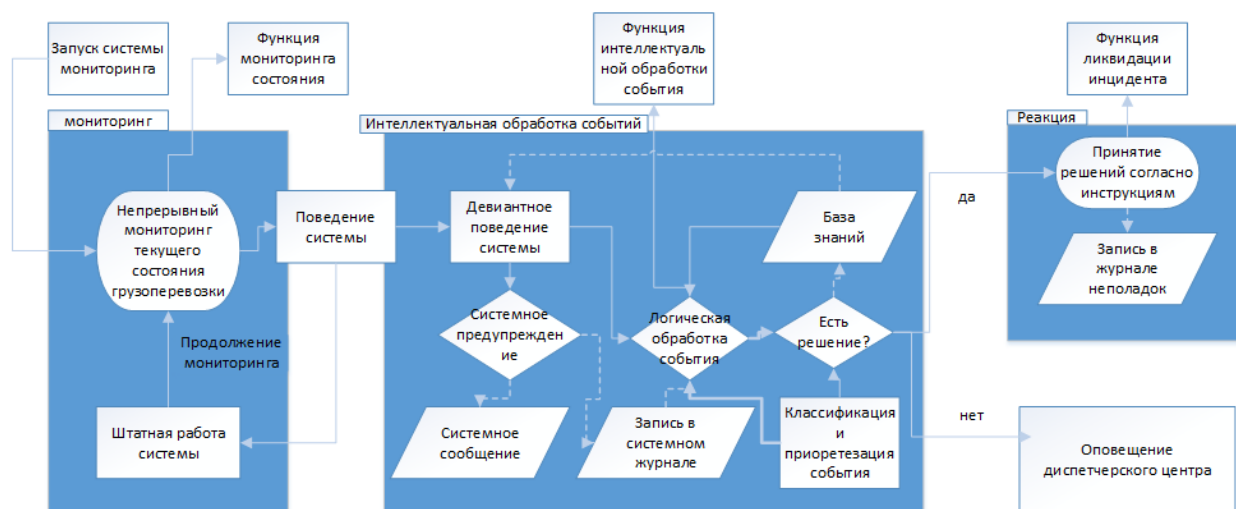


Рис. 19: Схема процесса мониторинга с применением мультиагентного подхода

Опишем поведение системы в виде бизнес-процесса с использованием Visio. Алгоритм поиска решений, в случае отклонения показателей от нормальных значений:

1. Некоторые параметры текущего состояния при мониторинге перевозки, наблюдаемые описанными выше агентами, выходят за допустимые пределы. Отклонение показателей является достаточным условием идентификации проблемы. Происходит идентификация и регистрация инцидента.
2. Начинается логическая обработка события, которая включает классификацию инцидента и его оценку.
3. Согласно алгоритму работы системы подбирается наиболее подходящее решение.
4. Если в случае поиска решения, выясняется, что сложившуюся ситуацию невозможно разрешить самостоятельно, то происходит информирование диспетчерского центра о невозможности соблюдения текущего расписания или маршрута движения.

5. Закрытие инцидента с проставлением даты.

Для общего понимания работы системы, небольшая иллюстрация программного и инфраструктурного архитектурных уровней решения, но подробное рассмотрение этих уровней не является целью данной работы, поэтому не будем на них останавливаться.

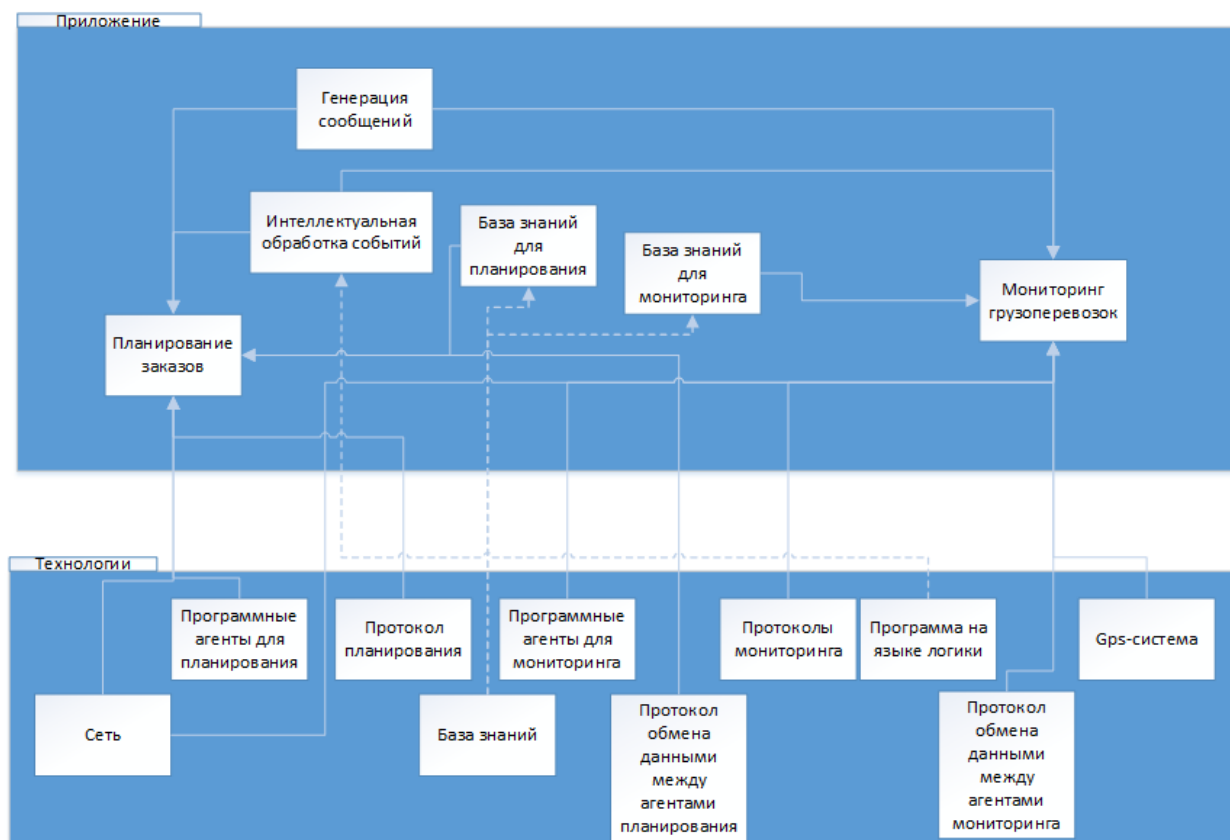


Рис. 20: Архитектура уровня приложений и инфраструктуры

3.8. Создание интерфейса для реализации системы мониторинга и управления грузоперевозками

3.8.1. Инструментальные средства

Интерфейс был написан с помощью локального веб-сервера XAMPP – кроссплатформенная сборка веб-сервера, содержащая Apache, MySQL, интерпретатор скриптов PHP, язык программирования Perl и большое количество дополнительных библиотек, позволяющих запустить полноценный веб-сервер. Использование XAMPP объясняется тем, что он

упрощает весь процесс установки и настройки инструментов, необходимых для написания приложения. Поэтому процесс разработки и тестирования веб-приложения получается очень удобным и приятным. В частности, мною были использованы следующие программные средства:

- HTML – стандартный язык разметки документов;
- CSS - формальный язык описания внешнего вида документа, написанного с использованием языка разметки;
- SQL – формальный не процедурный язык программирования, применяемый для создания, модификации и управления данными в произвольной реляционной базе данных, управляемой соответствующей системой управления базами данных;
- phpMyAdmin - веб-приложение с открытым кодом, написанное на языке PHP и представляющее собой веб-интерфейс для администрирования СУБД MySQL;
- PHP - скриптовый язык общего назначения, интенсивно применяемый для разработки веб-приложений;
- АТОМ – свободный текстовый редактор;
- Yandex браузер использовался для просмотра результатов.

3.8.2. Описание структуры базы данных и схема процесса грузоперевозки

Структура базы данных состоит из 10 таблиц:

- clients - содержит информацию о всех зарегистрированных пользователях, их логинах и паролях для входа в системы для совершения заказа;
- drivers - содержит информацию о именах водителей, работающих в компании;

- lorry - содержит информацию о имеющихся в компании грузовиках и их вместимость;
- distance - имеет вид матрицы расстояний между пунктами перевозок;
- newOrder - содержит информацию о: заказчике, водителе, машине, на которой будут перевозить заказ, вес груза, начальном пункте отправления и конечном пункте прибытия, дате поступления заказа;
- lorryCondition - содержит информацию о технических показателях машины, совершающей перевозку с номером из колонки order_ID;
- weatherHandling - содержит информацию о погодных условиях вдоль пути заказа с номером из колонки order_ID;
- parkingTime - содержит информацию о времени погрузки/разгрузки на складах для заказа с номером из колонки order_ID;
- driverCondition - содержит информацию о самочувствии водителя, совершающего перевозку с номером из колонки order_ID;
- speedControl - содержит информацию о скорости машины, совершающей перевозку с номером из колонки order_ID.

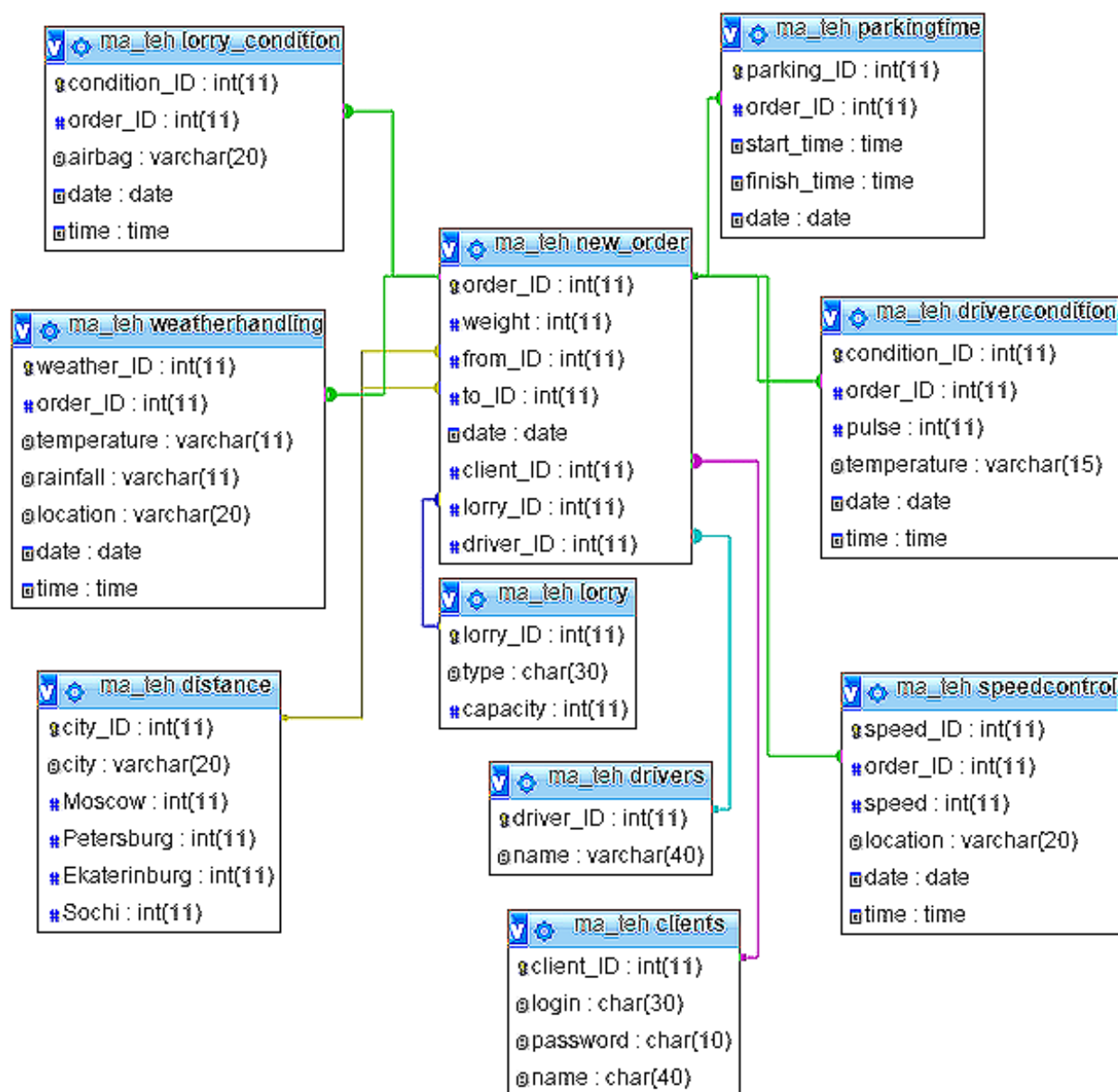


Рис. 21: Структура базы данных

По результатам второй главы и части третьей, можно объединить введённые ранее схемы процесса планирования расписания и мониторинга грузоперевозок (Рис.8 и Рис.19):

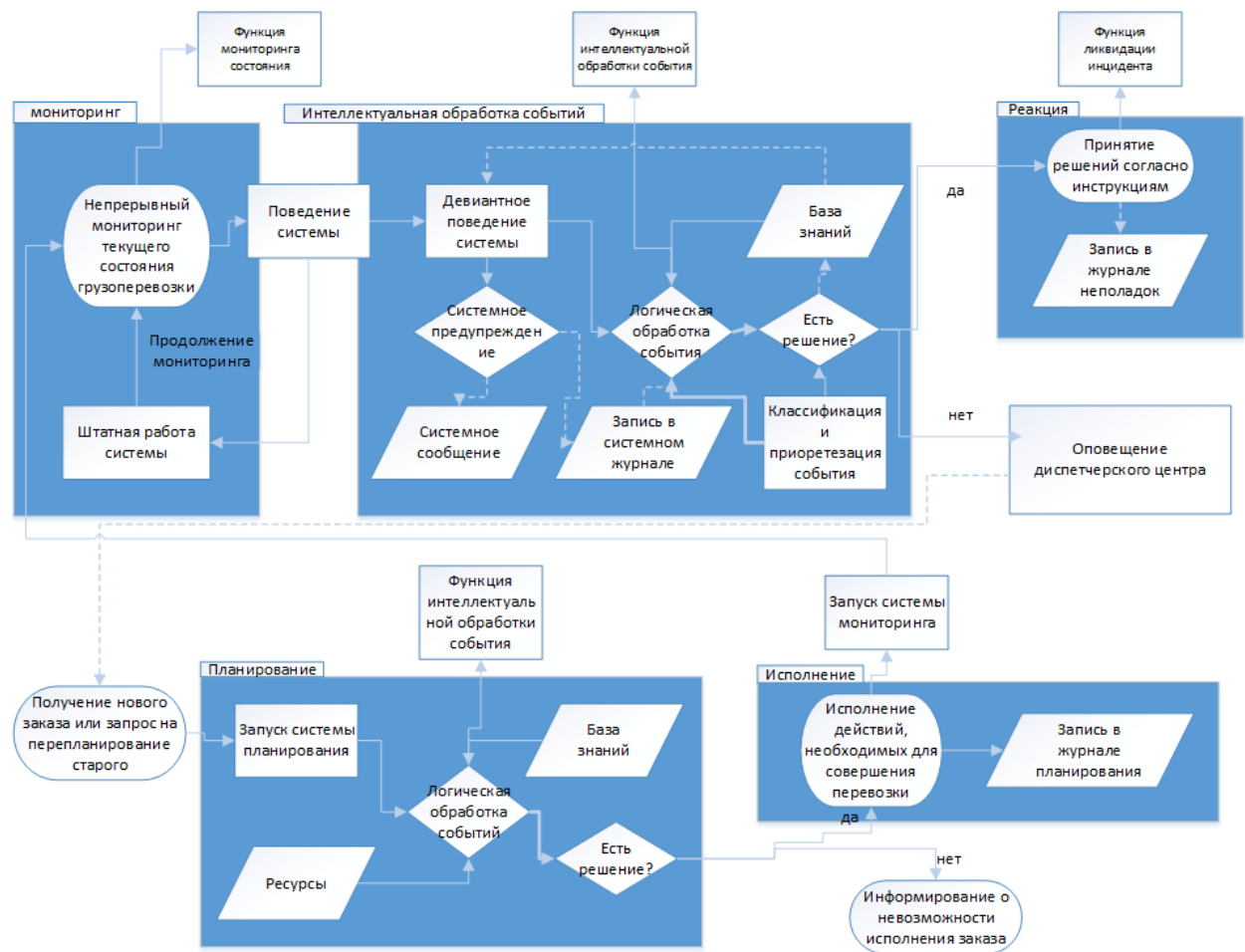


Рис. 22: Общая схема процесса грузоперевозки

3.8.3. Пример пользовательского интерфейса

Система представлена в виде web-приложения. На рисунке 23 представлена структура страниц.

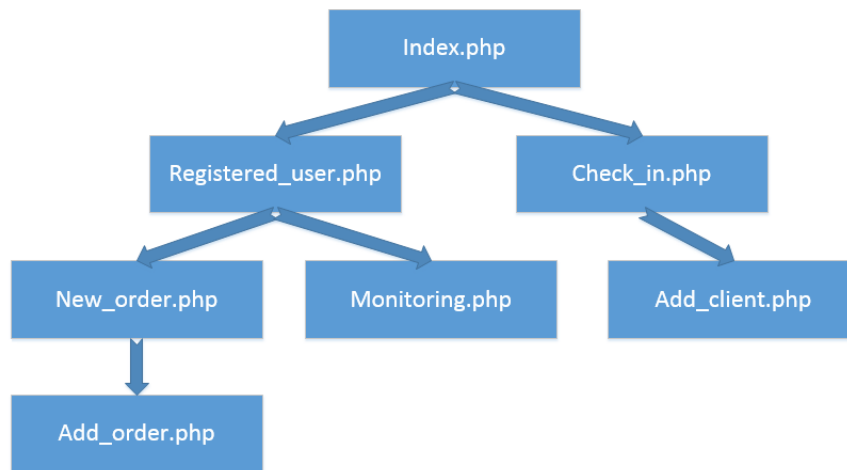


Рис. 23: Структура интерфейсных страниц системы

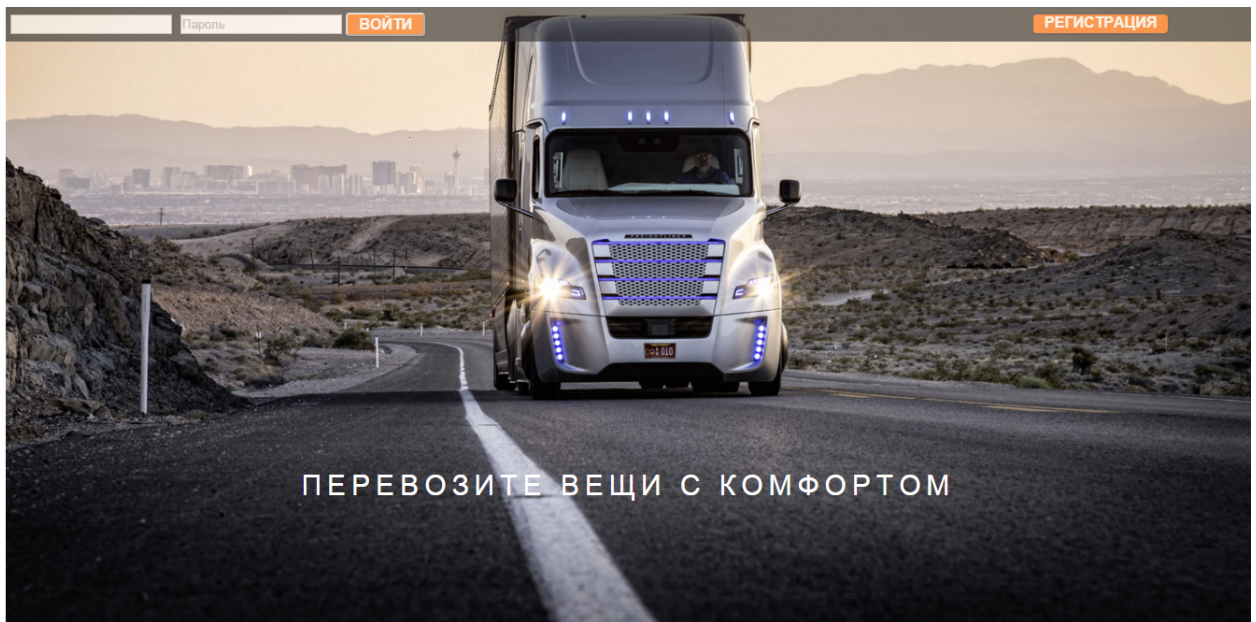


Рис. 24: Стартовая страница

На стартовой странице пользователю предлагается либо войти в систему с уже имеющейся учетной записью, либо зарегистрироваться.

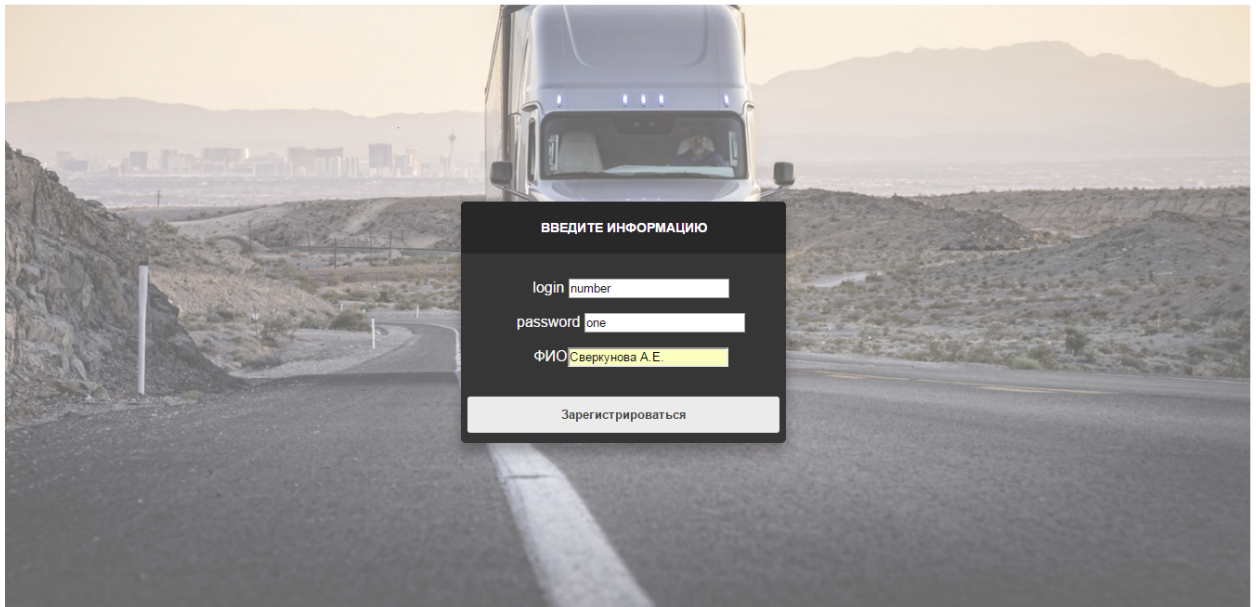


Рис. 25: Страница регистрации

Страница регистрации имеет минимальное число полей для заполнения, так как это демонстрационная система (Рис.25).

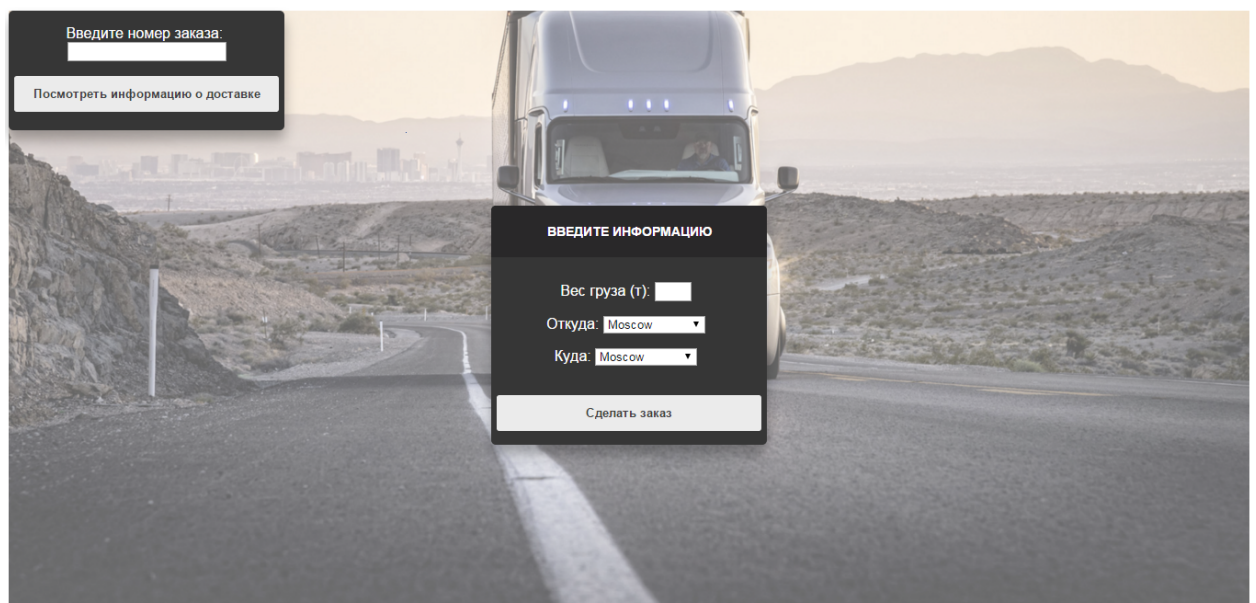


Рис. 26: Возможности для зарегистрированных пользователей

Форма для нового заказа имеет минимальное количество полей необходимых для совершения заказа (Рис.26). Начальный и конечные пункты непосредственно связаны с таблицей distance из базы данных, поэтому можно указывать только имеющиеся в базе пункты (это реализовано с помощью выпадающих списков).

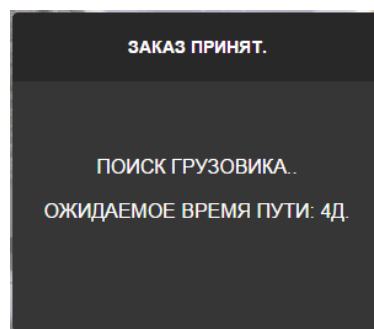


Рис. 27: Принятия заказа

При правильном вводе данных (то есть конечный и начальный пункты различны, а также не превышен максимальный вес груза, который равен 20т) появляется окно, подтверждающее о совершении заказа (Рис.27), а так же сообщается об ориентировочном сроке доставки, вычисленном с помощью таблицы distance, содержащей сведения о среднем количестве дней переезда из пункта в пункт.

Далее система должна составлять расписание и маршрут, составленные на основе мультиагентного подхода, описанного в главе 2.


Скорость	Местоположение	Дата	Время		Температура	Уровень осадков	Местоположение	Дата	Время	
66	(77, 20)	2016-03-15	14:00:00		+2	0 mm.	(77, 20)	2016-03-14	14:00:00	
60	(75, 24)	2016-03-15	15:00:00		+3	0,01 mm.	(75, 24)	2016-03-15	14:00:00	
63	(72, 27)	2016-03-15	16:00:00		+1	0 mm.	(72, 27)	2016-03-15	14:00:00	
64	(69, 29)	2016-03-15	17:00:00		+2	0 mm.	(69, 29)	2016-03-15	14:00:00	
0	(67, 32)	2016-03-15	18:00:00		0	0 mm.	(67, 32)	2016-03-15	14:00:00	
Начало погрузки/разгрузки		Конец погрузки/разгрузки			Дата					
13:00:00		13:50:00			2016-03-15					
17:52:00		18:19:00		2016-03-15						
Пульс	Температура	Дата	Время					Подушка безопасности	Дата	Время
65	36,6	2016-03-15	14:00:00					not used	2016-03-15	14:00:00
70	36,6	2016-03-15	15:00:00					not used	2016-03-15	15:00:00
68	36,6	2016-03-15	16:00:00					not used	2016-03-15	16:00:00
73	36,6	2016-03-15	17:00:00					not used	2016-03-15	17:00:00
67	36,6	2016-03-15	18:00:00					not used	2016-03-15	17:00:00

Рис. 28: Пример работы пятиагентной-системы для заказа №4 пользователя smirnov

Пользователю выводится информация о работе пяти-агентной системы по каждому из агентов в промежуток с 14:00 до 18:19 за 15 марта 2016. Заказ поступил в систему 15 марта 2016 (Рис.28).

Подушка безопасности	Дата	Время
not used	2016-03-21	10:00:00
not used	2016-03-21	10:30:00
not used	2016-03-21	11:00:00
used	2016-03-21	11:17:00

Рис. 29: Пример работы агента технического состояния автомобиля для заказа №5 пользователя smirnov

Пользователю выводится информация об использовании подушки безопасности в 11:17 21 марта (Рис.29).

Пульс	Температура	Дата	Время
71	36,7	2016-03-21	10:00:00
68	36,7	2016-03-21	10:30:00
70	36,7	2016-03-21	11:00:00
110	36,7	2016-03-21	11:16:54

Рис. 30: Пример работы агента, измеряющего показатели водителя

В 11:16:54 произошло резкое увеличение сердцебиения водителя. На основании показателей этой и предыдущей иллюстрации делается вывод о том, что грузовик попал в аварию. Об этом немедленно информируется диспетчерский центр, обязанный реагировать на чрезвычайные происшествия (Рис.30).

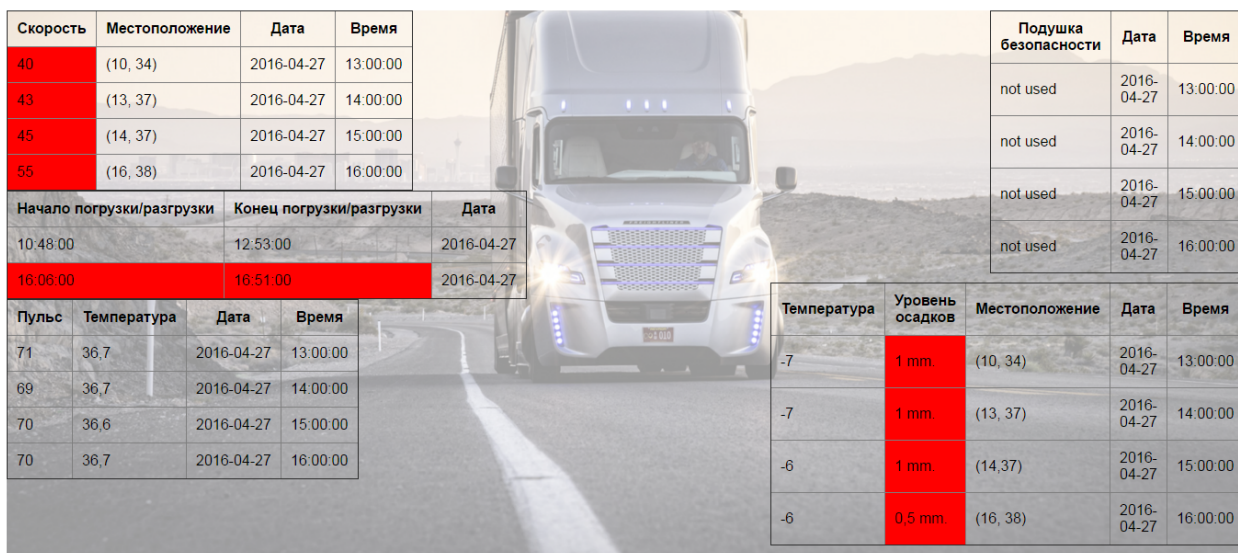


Рис. 31: Пример работы пятиагентной-системы для заказа №4 пользователя smirnov

На Рис.31 показан случай отставания от графика. По данным видно, что из-за высокого уровня осадков невозможно было двигаться с достаточной скоростью (скорость отмечена красным, так как её показатель был ниже границы разрешённой скорости для выполнения данного заказа). Поэтому время прибытия в пункт погрузки/разгрузки и сам процесс погрузки/разгрузки был совершён с опозданием (что отмечается красным цветом).

3.9. Выводы по третьей главе

В данной главе мной была формально описана система, состоящая из пяти агентов, описаны алгоритмы их работы. Затем, объединив эту разработанную систему с результатами второй главы, был разработан и проиллюстрирован в данной части интерфейс, для реализации мультиагентного подхода для задачи логистики. Рассмотрена и построена структура базы данных, состоящая из 10 таблиц, с которой связана MAS.

Заключение

Задача транспортировки различных грузов всегда будет актуальной: составление расписаний для перевозок, мониторинг текущих перевозок, корректировка маршрута или исправление отставаний от графика.

В работе для решения данной задачи мы рассмотрели возможность использования мультиагентного подхода для разработки системы поддержки эффективной транспортной логистики. Такой подход позволяет максимально автоматизировать систему с момента поступления заказа на перевозку и до момента доставки заказа в конечный пункт.

В начале работы были изучены и проанализированы теоретические подходы, необходимые для понимания устройства и работы мультиагентных систем.

Далее была рассмотрена возможность использования мультиагентного подхода для задач планирования заказов и мониторинга грузоперевозок.

Для задачи эффективного мониторинга процесса перевозки разработан алгоритм работы системы, состоящей из пяти агентов, в виде набора стандартных процедур.

Разработан прототип и интерфейсы системы для задачи транспортной логистики – мониторинга и управления процессом доставки грузов на основе мультиагентного подхода.

В завершение можно отметить, что использование мультиагентных технологий при решении логистических задач имеет большие перспективы.

Список литературы

[1] Амелина, Н. О. и др. Исследование моделей организации грузовых перевозок с применением мультиагентной системы для адаптивного планирования мобильных ресурсов в реальном времени // Пробл. управл. – 2011. - № 6. – с. 31–37.

[2] Бендерская, Е. Н. Мультиагентный подход в вычислительных технологиях: новые грани параллелизма и суперкомпьютинг / Е. Н. Бендерская, О. Н. Граничин, В. И. Кияев // Сборник научных статей 8-й Международной научной конференции «Информационные технологии в бизнесе». – СПб.: изд - во «Инфо-да», 2013. – с. 7–13.

[3] Виттих В. А., Скобелев П. О. Мультиагентные модели взаимодействия для построения сетей потребностей и возможностей в открытых системах // Автомат. и телемех. – 2003. - № 1. – с. 177–185.

[4] Граничин, О. Н. Информационные технологии в управлении. Учебное пособие / О. Н. Граничин, В. И. Кияев. – М.: Интернет - Университет Информационных технологий; БИНОМ. Лаборатория знаний, 2008. – 336 с.

[5] Граничина, Н. О. Мультиагентная система для распределения заказов // Управление большими системами. – 2010. – №30.1. - с. 549-556.

[6] Ерофеева В. А., Иванский Ю.В., Кияев В.И. Управление роём динамических объектов на базе мультиагентного подхода // Компьютерные инструменты в образовании. – 2015. - №6. – с. 36-44.

[7] Кияев В.И., Герасимов Р.В. Интеллектуальный CRM на базе мультиагентного подхода // В сб. «Стохастическая оптимизация в информатике». -СПб.: изд-во С.Петербург. ун-та, т. 8, вып. 1, 2012. - с. 50-94.

[8] Лекция 11. Транспортная логистика // Студми [Электронный ресурс]. – Режим доступа: http://studme.org/15100827/logistika/transportnaya_logistika. Дата обращения: 06.02.2016 г.

[9] Лемешева, Т. Л. Модели представления и обработки данных и знаний. Лабораторный практикум : учеб.-метод. пособие / Т. Л. Ле-

мешева, Н. А. Гулякина, А. И. Толкачев. – Минск: БГУИР, 2008. – 46 с.

[10] Логистика в малом бизнесе // Дистанционный консалтинг [Электронный ресурс]. – Режим доступа: <http://www.dist-cons.ru/modules/logistic/section8.html>. Дата обращения: 06.02.2016 г.

[11] Мультиагентные технологии и самоорганизация сетей связанных расписаний для управления ресурсами в реальном времени // Разумные решения, научно-производственная компания [Электронный ресурс]. – Режим доступа: [http://smartsolutions-123.ru/upload/medialibrary/e6/31_ИУМорские%20системы_№1\(7\)_2015_.pdf](http://smartsolutions-123.ru/upload/medialibrary/e6/31_ИУМорские%20системы_№1(7)_2015_.pdf). Дата обращения: 09.10.2015 г.

[12] Разработка приложений для мобильных интеллектуальных систем на платформе Intel Atom / К. С. Амелин [и др.]. – СПб.: Издательство ВВМ, 2012. – 2018 с.

[13] Скобелев П. О. Интеллектуальные системы управления ресурсами в реальном времени: принципы разработки, опыт промышленных внедрений и перспективы развития // Информационные технологии. – 2013. – с. 1–32.

[14] Скобелев П. О. и др. Мультиагентные технологии для управления распределением производственных ресурсов в реальном времени // Механика, управление и информатика. – 2011. – № 5. – с. 110–122.

[15] Татарников, О. Компьютерная диагностика автомобиля // Компьютер пресс [Электронный ресурс]. – Режим доступа: <http://compress.ru/article.aspx?id=12348>. Дата обращения: 09.02.2016 г.

[16] Шабунин, А. Б. и др. Разработка мультиагентной системы адаптивного управления ресурсами ОАО “РЖД” // Мехатроника, автоматизация, управление. – 2013. – № 1. – с. 23–29.